MANTECH Publications
International Journal of Software and Computer Science Engineering
Volume 4 Issue 2

Achieving Integrity Assurance of MapReduce in Cloud Computing

***Meenakshi*[*], *Ramachandra AC*[**], *Subhrajit Bhattacharya*[***]**
Assistant Professor[*], *Professor*[**, ***]
Dept. of Computer Science and Engineering[*, **], *Dept. of Data Science*
NitteMeenakshi Institute of Technology Bangalore[*, **], *Career Launcher, Bangalore*[***]
Corresponding author's email id:*kmeenarao@gmail.com*[*]
DOI:http://doi.org/10.5281/zenodo.3375498

Abstract

We are living in Internet world. Information or data is required on demand wherever, whenever required. Large amount of data in different formats which cannot be processed with traditional tools like Data Base Management system is termed as Big Data. MapReduce of Hadoop is one of the computing tools for Big Data Analytics. Cloud provides MapReduce as- a-service. In this paper we investigate and discuss challenges and requirements of MapReduce integrity in cloud. Review of some of the important integrity assurance frameworks are also focused with their capabilities and their future research directions. We discussed on algorithms for detecting collusive and non-collusive workers.

Keywords:*Cloud Computing, Hadoop, MapReduce, Security, Integrity*

INTRODUCTION

National Institute for standards and Technology defines Cloud Computing as " a pay-per-use model for enabling convenient, on-demand network access to a shared pool of configurable computing resources for example networks, servers, storage, applications and services, that can be rapidly provisioned and released with minimal management effort or service provider interaction". [1] Cloud computing is a mean of providing software, storage and processing as a service to the users for their applications with reasonable cost. Using cloud computing user can connect to applications over the internet. Pool of resources, elasticity and broad network access, accessibility, efficiency and measured services are some of the key features of cloud. It supports storage and

processing of data which could be fluctuating in terms of volume and velocity. All these feature of cloud computing made this as a tool of choice for big-data analytics. Cloud based big data analytics satisfy the requirement of complicated statistical analysis, linear regression, predictive analysis on big-data in the multidisciplinary fields like health-care, banking, business sectors, government projects, academia and many more.

Cost with respect to Hardware, software up gradation, maintenance or network configuration can be saved when cloud base big data analytics used. Enterprise can concentrate on analyzing data only, not the hardware or any other issues related to maintain it.

HADOOP- HDFS AND MAPREDUCE

Technologies behind cloud computing and big data a distinct and can operate mutually exclusively. Cloud has to play a big role in big data analytics. Cloud computing provides cost-effective solution to store large data set. Big data analytics can be made platform as a service with in a cloud environment. [2, 3, 4, 15,16] Hadoop can be thought as a platform that runs on cloud computing to provide us with distributed data mining due to the rate at which data are growing these days.

This chapter gives detailed explanation about Hadoop core storage component namely Hadoop Distributed File System and computing paradigm MapReduce.

Hadoop File system-HDFS

Hadoop Distributed File system is the default File system. Local File system, HFTP, F2, S3 and other mountable distributed file systems are also compatible with Hadoop Framework. Google File System is foundation for HDFS. It is designed work on thousands of commodity/cheaper reliable and fault tolerant machines.

In HDFS master/Slave architecture single Name node becomes a master node and N, N>=1 number of data nodes becomes Slave nodes. Metadata and actual data are stored in master and slave node respectively.

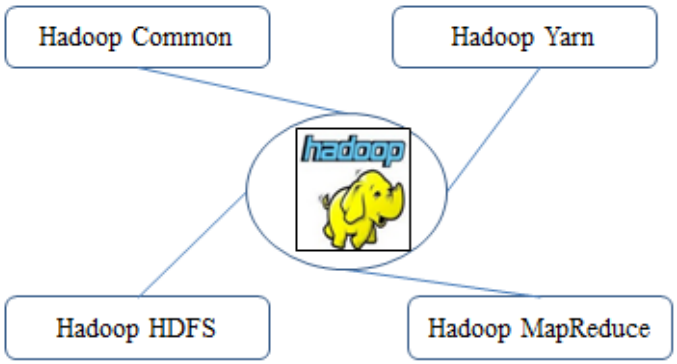

Figure: - 1 Main Modules of Hadoop-Hadoop common, Hadoop Yarn, HDFS and MapReduce constitute Hadoop Framework.

- Determining mapping of fragmented input files to the slave nodes in terms of fixed block size; 64KB by default.

- Instructing data nodes for block creation/deletion/replication.
  Replication factor is three by default.

- Making data nodes to perform read/write operations. HDFS provides commands to interact with file system which are similar to those of UNIX shell commands.

How Does Hadoop Work?

The first stage is Submitting job, which allow user/an application to submit a job/task to be processed to the Hadoop's Job-client. To do so, location of the input and output files is specified by the user. And also provide the java classes in the java files which comprise of computing algorithms namely Map and Reduce. Different parameters are also given to the job which does Job configuration. From the JobClient, these jar executable files and input configuration goes to the JobTracker.

Tasks to be handled by JobTracker are as follows:

- Assigning jar files and configuration to slaves/TaskTracker.
- Scheduling tasks
- Monitoring tasks.
- Report about diagnostic/status information to the JobClient.

Last third stage is execution phase where tasks are executed by TaskTracker on different nodes as per the MapReduce code written by the user.

Hadoop MapReduce

MapReduce comprises two components namely Map task and Reduce task (Fig.4).

Map Task after taking input data converts individual elements into key/value pairs. This output becomes input to the next reduce task. It combines these data tuples into a smaller set of tuples [10]. By default both input and output are saved in HDFS file system.

MapReduce is highly scalable over thousands of nodes. Data processing application has to be decomposed into Mapper and Reducer once in the beginning, and same will be applied to data residing in multiple machines. This is the advantage of MapReduce strategy. Key idea is sending machine to where the data resides. Map, Shuffle and Reduce are three major stages of algorithm. Map or the Mapper takes its input line by line from HDFS by default and processes it by creating small blocks of data. Next stage is combination of Shuffle and Reduce phase. New set of output goes back to the HDFS. During processing MapReduce is sent to the specified servers in the cluster. Cluster collects individual results and reduces these into final result form and sends to the HDFS server.

Executing the task

Input to the Mapper has to be converted as pair of <Key,Value>. Output also is in this form only. Map function is applied to all pairs. Zero or more intermediate (1key, 1value) pairs are output of map function. Next stage is grouping of pairs based on 1k value of each pair. Reducer is called for each such group. Output is final result. MasterNode/NameNode is the Node where JobTracker runs and which accepts job requests from clients. JobTracker works with the name node which is single master node. It manages all Hadoop resources. It keeps track of resource consumption and also schedules the master and slave jobs to appropriate machines.

SlaveNode/DataNode is where Map and Reduce programs runs. TaskTracker runs in SlaveNodes. Job of TaskTracker is monitoring tasks assigned to SlaveNodes and re-executing the failed tasks are also additional responsibilities of it.

Word Count Example

One of the well-known illustrations to understand MapReduce is word count program. It is depicted in Fig.5. In the newly setup infrastructure of Hadoop-based big-data ecosystem, many questions need to be answered. These questions are about security of Hadoop ecosystem, security of data residing in Hadoop, secure manner of accessing Hadoop ecosystem, the way to enforce security models and many more.

MAPREDUCE SECURITY THREATS AND PRIVACY CHALLENGES

When cloud is providing MapReduce as a service, it has to deal with new security and privacy issues. Some of such challenges are explained here. MapReduce is dealing with Big Data which is large and arriving at high speed from various inputs. Single system hold single copy of data in one location only where as in MapReduce, replicated splits of data need to be transferred and stored securely. Cloud alone may not have distributed computing for every task. But MapReduce means computing distributed replicated chunk of data. It needs to secure both distributed nodes and replicated data. Attack may yield wrong output from effected mapper or reducer, modify the data, transfer data to third party etc. data flow occur between clouds, or between storage nodes or between computing nodes. Adding security and privacy mechanisms should not be a burden on MapReduce working efficiency.

When MapReduce is deployed in public cloud, it is more vulnerable to security threats as compared to private deployment. Authentication, authorization and access control" are very essential requirements for MapReduce computational nodes. Authorization is the process of identification of adversarial mapper or reducer or user. After successful authentication, access privileges of mappers and reducers are checked so that they can proceed to access and framework. "Availability" of data, mappers and reducers are always for authenticated and authorized users without much delay. Strategy provides effective solution for detecting malicious workers, but difficult to identify when all replicated tasks are handled by collusive group. Table I gives brief about MapReduce security threats.

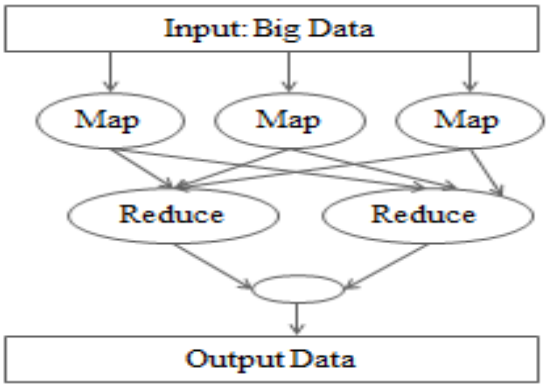

Figure:-2 Hadoop- Map and Reduce algorithm Overview

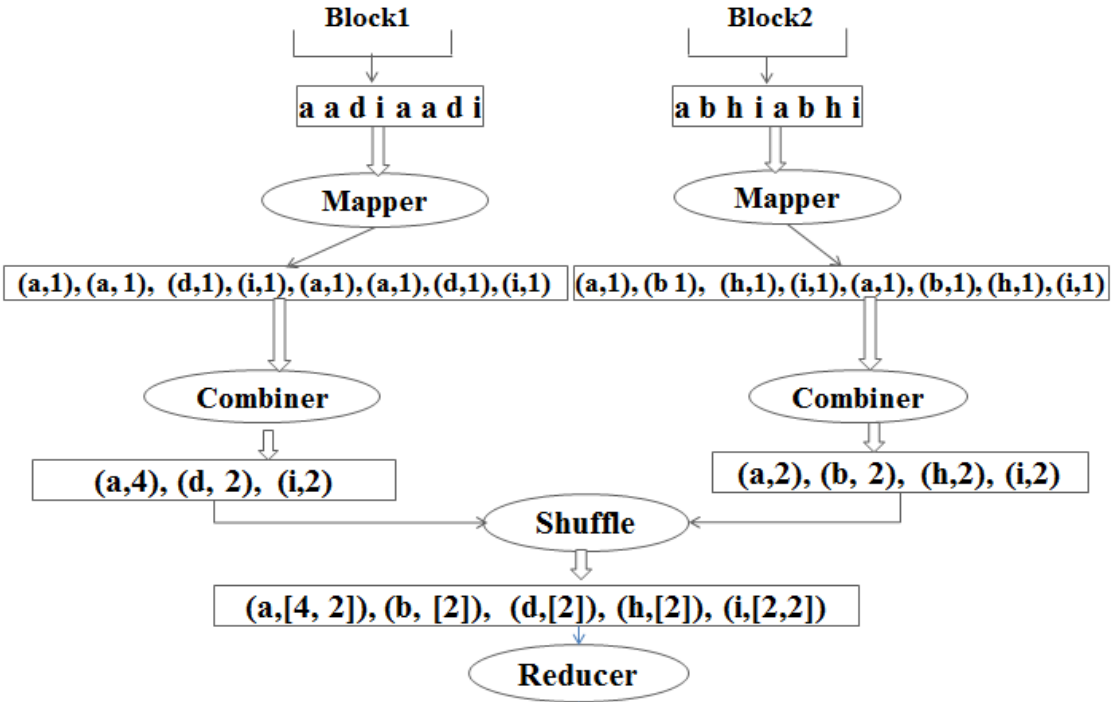

Figure:-3 Work Flow of MapReduce Algorithm in word count task for given file fragmented into block1 and block2.

RELATED WORK ON INTEGRITY ASSURANCE IN MAPREDUCE

"Data Integrity is the assurance that data received are exactly as sent by an authorized entity that means without containing modification, insertion, deletion, or replay".

In SecureMR framework [5], master is assumed to be safe and workers are not trusted. Distributed File System DFS incorporated with integrity assurance so that workers are provided with integrity protected data. "Each worker is having public/private key pair and any worker can generate and verify signatures and no worker can forge other's signatures". Along with master node, intermediate results obtained from two different map reducers of same replicated task is checked for consistency by other worker also. It provides scalability and efficiency. Commitment protocol and verification protocol are implemented to provide security for MapReduce tasks. Future research direction is applying sampling technique to find inconsistency and provide integrity when all duplicated tasks are processes by collusive attackers. [5] Strategy provides effective solution for detecting malicious workers, but difficult to identify when all replicated tasks are handled by collusive group.

Table:-1Various Attacks on MapReduce

	Attack	Passive	Active	Definition	Effect on MapReduce	Attack on
1	Impersonate attack		√	Attacker pretends to be a legal user by gaining passwords or weak encryption schemes with brute force attack	Effected legal user sometimes may be charged for using cloud. Attacker may do data leakage or wrong computations	Authentication
2	Denial-of-Service (DoS)		√	Attacker causes system non-functional. In MapReduce context system means nodes, or mappers or reducers.	Attacked node may cause other working node non-functional by sending repeated execution request. DoS cause heavy network traffic.	Availability of data, mapper, reducer
3	Replay attack		√	Adversary resends the valid message to mappers or reducers	Make node busy. It may replay authentication details and cause impersonate and DoS.	Authorization
4	Eavesdropping	√		Observing inputs, outputs and intermediate results of nodes without knowledge of computing owner	Adversary gain knowledge of intensive computations.	Confidentiality computations and data
5	Man-in-the-middle attack		√	Attacker modifies the data of communication between two nodes	It may lead to DoS, impersonation or replay.	Confidentiality computations and data
6	Repudiation	√		Node falsely denies execution request.	Mapper or reducer falsely denies the execution request of already accomplished task.	Authorization, Authentication

Paper [6] presents "integrity framework for both collusive and non-collusive mappers". This framework assumes both storage and masters are trusted. Mappers and special limited verifier workers are executing on trusted node but mappers are not trusted. It is based on replicating each mapper which identifies non-collusive mappers. It computes result without contacting any other non-collusive nodes. Next is identifying collusive nodes, which communicate with other malicious node in order to send same type of output.

It is very tedious job to identify collusive nodes. Dedicated verification nodes send quiz-type verification to each computing node which verifies portion of result randomly. Whichever node fails to answer quiz is considered as malicious node. Each mapper worker prepares intermediate result as well as MD5 hash code of this result. These are cached for obtaining result of replicated task. Once both workers clear k quizzes. Future research direction is making this framework worth in case of untrusted reducer workers. Second possible enhancement to this work is reusing verification node computation of reused tasks which reduces its workload. Table2 gives detailed comparison of some of the algorithms on assuring integrity in MapReduce.

Table 2. MapReduce Integrity Assurance Frameworks

	Framework	Verification Schemes	Attack Model	Concept	Future research directions
1	"VC3: Trustworthy Data Analytics in the Cloud using SGX" [7]	Hardware / checkpoint based	Physical processors ensure integrity of memory region of systems.	User uploads encrypted MapReduce code to work on encrypted file. Key exchange protocol is executed to decrypt MapReduce inside workers. Result is again encrypted.	Tampering Processor packages, DoS attack, traffic analysis, fault injections need to be addressed.

2	"TMR: Towards a Trusted MapReduce Infrastructure" [8]	Hardware / checkpoint based	Computing infrastructure is trusted. Master can be verified periodically by third party.	Trusted Platform hardware Module does remote attestation of workers and programs loaded to workers are checked for reliability.	Framework can be verified for large scale infrastructure and real-life practical workloads.
3	"Towards Trusted Services: Result verification schemes for mapreduce" [9]	Watermarking / sampling based	Workers of MapReduce cannot decide whether input is having injected data or not.	First data is preprocessed by verifier with injected watermark. It verifies the correctness of MapReduce result. Random sampling also applied.	Instead only "text-intensive task", "numerical data-intensive" tasks also need to be considered.
4	"Viaf: Verification-based integrity assurance framework for mapreduce" [5]	Watermarking / sampling based	Both storage and masters are trusted	Dedicated verification nodes send quiz-type verification to each computing node which verifies portion of result randomly. Whichever node fails to answer quiz is considered as malicious node.	Making this framework worth in case of untrusted reducer workers and reusing verification node computation of reused tasks which reduces its workload.
5	"A Result Veri1fication Scheme for MapReduce	Watermarking / sampling based	Master is trusted and computation providers are	In prepressing, secondary cluster is formed with equal range of data to every mapper. Small fraction of total number	Some workers may not being verified at all. Randomized worker selection can be improved further

	” [10]		malicious without disturbing accuracy level of output.	of workers is selected for verification.	for complex computations.
6	“Securemr: A service integrity assurance framework for mapreduce” [6]	Replicatio n/ double check based	Only master is trusted. All workers are in untrusted domain.	Commitment protocol and verification protocol are implemented to provide security for MapReduce tasks. Along with master node, intermediate results obtained from two different map reducers of same replicated task is checked for consistency by other worker also.	Sampling technique to be applied to find inconsistency when all duplicated tasks are processes by collusive attackers.
7	Distributed Results Checking for MapReduce in Volunteer Computing [11]	Replicatio n/ double check based	Master is trusted and workers are not. All workers are non-collusive independe nt.	Design of the “result certification” in MapReduce computation in Desktop Grid has been addressed using “Majority Voting method”. Verification is decentralized involving not only master but workers also. In Naming scheme workers attach a key to the result computed which helps reducers to check the	Collusive workers also may be present in the system and produce erroneous results. Framework can be redesigned t address this issue.

			origin of result.	
8	"Achieving accountable MapReduce in cloud computing"[12]	Replication/ double check based	Cloud data is correct. Workers are unaware of existence of auditors who replay the task. Auditor Group is not malicious actions. Workers cannot reclaim till completion.	It forces each machine to be held responsible for its behavior. Accountability test is done by auditors which check all machines and detect malicious nodes. Auditor replays the task and compares result with original result. Probability-accountability reduces the number of records to be checked.	If more than one auditor is involve in testing A's task, computational task increases. Master need to verify the correctness of idle worker to select as an auditor.

The Authors of [13] HuseyinUlusoy and others developed a novel framework for computation Integrity problem of MapReduce based on replication scheme. In [14] Yan Ding and others proposed a framework for protecting MapReduce against collusive attackers and assuring integrity without extra re-computations. Analysis of "undirected Integrity Attestation Graph" helps to identify both collusive and non-collusive attackers. Assumption is both master and reducers are in trusted domain whereas mappers are untrusty. To verify workers' trust in terms of consistency of mappers' results, edges of graphs have marked either zero or one. Zero indicates both mappers have given different intermediate results but otherwise no. Earlier case is termed as inconsistency pair of workers and later one as consistency pair.

In [14] Y Ding and others proposed a scheme to detect attacks in both map phase

and reduce phase and is based on not replication based. First assumption of attack model-nodes are communicating in cryptographically secured way and only master node is assumed to be trusted. It is probe injection based verification method to achieve integrity of MapReduce computations and also to identify malicious workers. "A probe consists of data injected into the original input dataset. The result of the probe set can be pre-computed before the entire input dataset is processed in the MapReduce system. A probe has two attributes: data value and location. The data value is the specific value in the computation; the location is its position in the entire dataset after injection."

CONCLUSION

MapReduce became a fault-tolerant, efficient, and scalable data processing tool for large datasets. But when MapReduce introduced over public and hybrid cloud computing, addressing security and privacy became important concerns. Since then many algorithms and frameworks are coming into picture to address these sensitive issues. Some of the important algorithms and frameworks had been surveyed here in details. Comparison table is also presented. Based on survey we listed these frameworks under three categories namely hardware/check based, sapling bases and finally replication based. During this survey we observed that lot of research need to be done in the area of security issues in file system, trust on hardware etc. No much work is shown where master is malicious or in case of untrusted cloud provider.

Two or more of these algorithms can be integrated to provide more sophisticated secured BigData-MapReduce Analytics model in Cloud.

REFERENCES

I. AjithBailakare, and Meenakshi, "An Introduction to Cloud Computing and its Security Issues and Challenges - A Literature Review", IJEECS, Vol. 6, Issue 5, 2017.

II. Han hu, Yonggangwen,Ttat-sengchua, and Xuelong li, "Toward scalable systems for big data analytics: A Technology Tutorial", IEEE transactions, vol. 2, Pp. 652-687, 2014.

III. RaghavendraKune, Pramod Kumar Konugurthi, ArunAgarwalRaghavendraRaoChillarige and RajkumarBuyya, "The

anatomy of big data computing", Wiley Online Library, 2015.

IV. Shankar Ganesh Manikandan and Siddarth Ravi, "Big Data Analysis Using Apache Hadoop", IEEEE Explore, International Conference on IT Convergence and Security, Pp. 1-4, Oct. 2014.

V. W. Wei, J. Du, T. Yu, and X. Gu, "SecureMR: A service integrity assurance framework for mapreduce," in ACSAC, 2009, pp. 73–82.

VI. Y. Wang and J. Wei, "VIAF: verification-based integrity assurance framework for MapReduce", International Conference on Cloud Computing, Washington, DC, USA, Pp. 300–307, 2011.

VII. F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "Vc 3: Trustworthy data analytics in the cloud", IEEE Symposium on Security and Privacy, Vol. 15, 2014.

VIII. A. Ruan and A. Martin, "TMR: Towards a trusted mapreduce infrastructure," World Congress, Pp. 141–148, 2012.

IX. Huang Chu, Zhu Sencun and Wu.Dinghao, "Towards Trusted Services: Result Verification Schemes for MapReduce", International Symposium on Cluster, Cloud and Grid Computing 2012.

X. Pareek G., Goyal C. and Nayal M., "A Result Verification Scheme for MapReduce Having Untrusted Participants", Intelligent Distributed Computing, Advances in Intelligent Systems and Computing, Vol 321, 2015.

XI. M. Moca, G. C. Silaghi and G. Fedak, "Distributed Results Checking for MapReduce in Volunteer Computing", International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, Shanghai, Pp. 1847-1854, 2011.

XII. Zhifeng Xiao, Yang Xiao, "Achieving Accountable MapReduce in cloud computing",

Future Generation Computer Systems, Pp. 1-13, 2014.

XIII. HuseyinUlusoy, Murat Kantarcioglu, ErmanPattuk and LalanaKagal, "AccountableMR: Toward Ac

XIV. countable MapReduce systems", International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, Pp. 451–460, 2015.

XV. Yan Ding, Huaimin Wang, Songzheng Chen, Xiaodong Tang, Hongyi Fu and Peichang Shi, "PIIM: Method of Identifying Malicious Workers in the MapReduce System with an Open Environment", International Symposium on Service Oriented System Engineering, 2014.

XVI. Meenakshi, A. C. Ramachandra, M. N. Thippeswamy, and AjithBailakare, "Role of Hadoop in Big Data Handling", ICICI 2018, LNDECT 26, Pp. 482–491, 2019.

XVII. R. Chandana, D. Harshitha, Meenakshi, and A. C. Ramachandra, "Big Data Migration and Sentiment Analysis of Real Time Events Using Hadoop Ecosystem", ICICI 2018, LNDECT 26, Pp. 1–7, 2019.

Cite this Article As

Meenakshi, Ramachandra AC, Subhrajit Bhattacharya (2019) **Achieving Integrity Assurance of MapReduce in Cloud Computing** International Journal of Software and Computer Science Engineering, 4 (2), 20- 33

http://doi.org/10.5281/zenodo.3375498